Weaving The Web Into Legacy Information Systems

Roy Schoenberg, Larry Nathanson, Charles Safran, Daniel Z. Sands
Center for Clinical Computing, Beth Israel Deaconess Medical Center
Harvard Medical School

ABSTRACT - For many years, client-server systems were developed as the backbone of clinical computing in leading hospitals around the country. Beth Israel Deaconess Medical Center now faces the challenge of bridging the technology gap between such systems and the Internet. While developing Web interfaces to legacy clinical systems gives a taste of the future, it is clear that complete institutional migration to the Web is not imminent. Asking clinicians to utilize two different systems, Web-based and legacy, in the interim phase is just one of the difficulties in such transition. This paper describes "Mbridge", a solution that allows legacy system users to exploit the benefits of the Internet in a fashion that does not interfere with their workflow and is both simple and affordable to implement. The service allows clinicians to work on the legacy platform while context-sensitive clinical content is streamed to the browser without their intervention. Using the system, we can gradually expose clinicians to new Web-based applications and resources without forcing them to operate two computing environments simultaneously. The service achieves these goals by means of linkage and coordination rather than by code-translation, data exchange or replication.

Introduction - Migrating clinical systems to the Web seems to be on the agenda of every health system CEO and a top challenge for ambitious CIO's[1]. Other than seeming to be the "way of the future" in terms of delivery and retrieval of information[2], such migration now enjoys wider acceptance by care providers who progressively feel more comfortable with the conventions used by Web-based interfaces[3]. Nevertheless, an institutional transition to the Web poses numerous challenges beyond code writing[4]. Trusted legacy systems have been developed over a lengthy period of time, tested and tweaked to satisfy provider requirements. Providing lesser functionality on even the most promising platform is likely to be rejected by users. This translates to the impossible task of implementing a Web-based system that replaces the full functionality of the legacy system from the start. In reality, partial rollouts that co-exist with legacy system application are likely to be more common[5]. Care providers, who now need to operate both systems interchangeably, justly complain that this approach disrupts their workflow. In many cases, providers who initially support the transition, turn to criticize it when they realize the magnitude and duration of the interference involved in the transition. It comes as no surprise, that much of the effort invested in deploying Web-based systems is directed to alleviate the interim phase when the two generations of systems co-exist[6]. Such effort may be as minimal as to provide encouraging information about what can be expected down the road, or as intensive as building intermediate applications for the sole purpose of maintaining use of the system[7]. The system we present provides a bridge for users during this transitional phase offering them access to Web functionality seemingly from within legacy systems. This allows administrators to progressively shift functionality to Web applications as they become available. By linking the two platforms we deliver a "feel" of an integrated system[8] at the time when both systems independently deliver separate, required functionality.

The need to migrate to the Web – Although somewhat rhetorical, comparing the cost of development and implementation to the return-on-investment may not always justify migration endeavors. In many cases, such investment (for a specific functionality) will not generate a return until other computerized services (e.g., billing, clinical databases) are modified to support connectivity[9] and data transfer[10]. Before we suggest a remedy to the illnesses of transition, we must first consider its final objectives and how our tactics will support the long-term organizational strategy. Here, we must separate the objectives of the different parties involved, namely the users and the organization.

What do end-users want - End-users objectives typically revolve around interface and content. Beyond the intuitiveness provided by the graphical user interface, the use of pictures, animation, sound, video, color and other visual aids, may all increase the information that can be comprehended in a

1067—5027/00/$5.00 © 2000 AMIA, Inc.

single glance (page). This, in addition to the spectrum of data objects that can now be delivered (e.g. radiographic images, electrocardiograms, pathology specimens etc.). Clinical content on the Web is another justification[11]. Beyond the use of Medline, providers can take advantage of clinical guidelines, drug monographs and even concise up-to-date treatment protocols that may be helpful during patient encounters[12]. Clinical sites now provide services that can be used during the clinical workflow ranging from clinical formula calculation to complex case management decision support tools. Many non-clinical utilities can also assist the care provider. Typical examples include Web-based scheduling and personal information managers. The attempt to deliver full application functionality via the Web (the ASP - Application Service Provider model) also stresses the need to integrate and migrate the provider's desktop to the Web. Another promising aspect of Web integration is patient-provider interaction[13,14]. Such interaction can be moderated via secured, web-based email or utilize comprehensive interactive applications that perform patient intake and monitoring from their homes, reducing the need for routine visits in certain subsets of patients[15] (e.g. diabetic, asthmatic).

What does the organization need - From the organizational perspective, the thin client model holds promise in terms of cost of implementation and administration[16]. Scalability of Internet/Intranet systems is, arguably, one of their most compelling features. The introduction of XML has enhanced the functionality of the Web from a window to present Internet-based information to one that handles data transactions[17]. The upcoming release of HL7 version 3 is a good example of this transformation. Remote access to clinical information, the tentative gold mine in healthcare Internet, remains an attractive argument for migration but may lose clout as VPN (virtual private networks) become integral part of some operating systems. Security issues seem to be an argument against migration considering the vulnerability involved in opening Web communication channels. Nevertheless, when used correctly[18], the Web still offers better security for information delivery when compared to the alternatives commonly used in its absence – fax, email, regular mail and telephone. Although it is not the purpose of this paper to examine the reasons for undertaking Web migration, the above considerations are crucial when outlining the objectives and methodology of such a project.

Challenges to Web migration – in addition to the straightforward challenges in building a comprehensive, robust Web-based information system, one challenge is typically underestimated. This challenge results from the fact that the new systems do not fill a void in many organizations but rather occupy the space of an existing, operational clinical system. Unlike the legacy system, which was build over decades, the new system is often unrealistically expected to be better right out of the box. The difficulty in "fitting into the shoes" of the legacy system is often wrongfully dismissed by emphasizing the new functionality offered by the Web-based system.

In reality, users prefer familiar interfaces even at the price of lesser functionality. This is especially true for new, additional functionality that is, by definition, not felt to be necessary. The learning curve required from users cannot be justified by the benefit to the organization, and intolerance to version 1.0 glitches cannot be handled by reiterating the vision (as often happens). Moreover, some of the functions in legacy systems cannot (or should not) be migrated at all. This can be the result of the strategic design of the Web system or simply because the process is too expensive. Examples include migration of knowledge that is rendered as code rather than data (e.g. alarms, reminders, medical logic modules, some interactive applications) or the effort necessary to convert old data archives in proprietary structures[19] (e.g. "Mumps globals") into "Web-friendly" databases.

As a result, migration to the Web is usually accomplished in a gradual and staged manner rather than as a single rollout. Consequently, parts of the overall required functionality will be shared by the two systems, often in a redundant manner. Users will be asked to use both systems simultaneously to access the functionality they need. Transition to the new system will be governed by the speed legacy system features are replicated in the Web platform and are announced to users. This phase of dual operation may extend, as our experience shows, over considerable time measured in years. Maintaining the compliance and cooperation of users as well as the momentum of the project at this time is crucial to the eventual success of the implementation. We thus turn to discuss "Mbridge" – a system that connects the legacy and Web-based riverbanks and simplifies the journey between them.

Design requirements – Three key requirements directed the design process of the system. Primarily, the system itself should be transparent to users to avoid workflow disruption. Secondly, the system should allow users who choose to work with the legacy interface to continue doing so until they feel comfortable to make the switch. Thirdly, no investment should be made (financial, manpower, programming, testing or support) to make the legacy system compatible with the intervention. Minimal coding should be required on the legacy system side. These requirements dictated an architecture that will interact with the legacy system as a service rather than as a feature. A demanding solution such as encapsulating browser functionality into the legacy system was therefore out of the question.

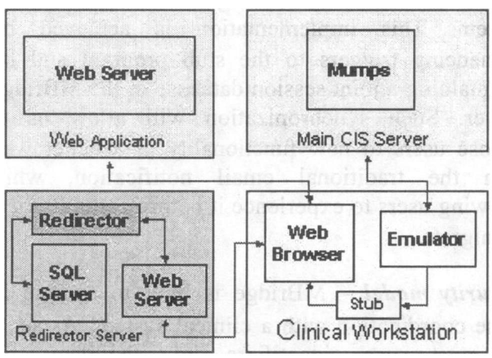

Figure 1: The "MBridge" Architecture

System architecture (Figure 1) – to understand the mechanism of the service, we must first consider the system it serves. The backbone server (Mumps based, in our case) interacts with users via a standard emulator that lives on the client machine (running Microsoft Windows™). The communication between the emulator and the server can take several forms including Telnet or direct modem dial-up. The emulator has a limited ability to interact with its immediate environment on the client machine via executing a command line argument sent by the server. In practice, the emulator can run an executable that resides in its own directory on the client hard drive. The emulator cannot interact with the triggered executable thereafter nor monitor its execution. This model of emulator functionality is found in many client-server systems and does not depend on the type of server to which the emulator connects. It is important to realize that whereas our legacy system is based on Mumps, the architecture is applicable to any emulator-based client-server system.

At the heart of the system lies a 28kb executable file (nicknamed 'stub') that is added to the emulator directory on the client machine (either manually or by remote administration techniques). As the stub does not register or utilize DLL components, no installation is required. The local emulator can therefore call the stub when the appropriate signal is dispatched from the main server. The signal sent contains a string, which corresponds to a fixed URL with variable arguments. The function of stub is threefold - identify the location and type of the installed browser, launch it and send a URL to it (the string signal sent by the server). In our implementation, this URL will direct the browser to a dedicated "redirector server", another component of the system, together with the list of query-string arguments to moderate its operation. The redirector, operating a SQL database, uses the arguments to identify and construct another URL that addresses internal or external Web resources. The new URL is then sent back to the client browser (as a simple redirect), which queries the resource server and retrieves the desired content. It is the responsibility of the redirector to append any parameters and values (e.g. passwords, query modifiers) onto the URL to allow interaction with the target applications. The correct interaction protocol with each target URL is kept as data in the database. The redirector is also responsible for converting data arriving from the legacy system into the format understood by the target application. In order to keep the amount of data going through stub to a minimum, much of the fixed information that is required by the target sites is stored in the redirector and is added in the final stage to the target URL, prior to redirection.

The redirector database consists of several linked tables of which three deserve attention. The "URL" table contains the base URLs of resources we are interested in exposing (e.g., a commercial drug database). The "parameters" table contains as many of the operational parameters required by the resource as needed (e.g., search_name, drug_form etc.). The "values" table contains the default values that should be plugged into the operational parameters if their values are not supplied in the request coming from the legacy system. This allows the redirector to build the full URL given as little or as much information as is supplied by the legacy system in each interaction instance.

Another component in the system is the "moderator". This component is responsible for moderating the interaction with applications that do not accept explicit operands using query-strings.

Such applications require the use of full-page submissions (e.g. posting forms with hidden fields) to deliver their functionality. To achieve that, the moderator dynamically constructs an HTML image of such a page, plugs in the desired values and sends it to the client's browser as the response to Stub's request. The generated pages are configured to submit upon loading so the client browser is immediately returned with the desired response from the target resource.

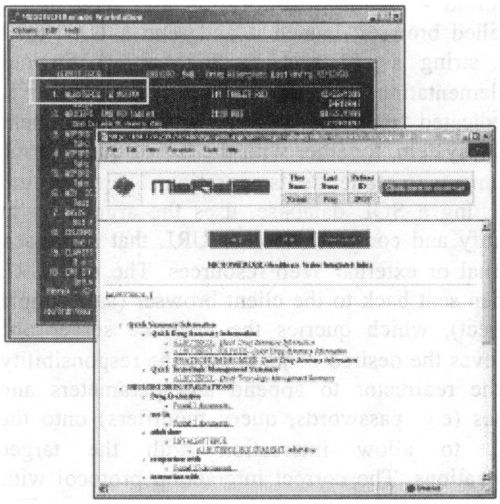

Figure 2: Prompting Drug Information

Workflow methodology – using MBridge we are now capable of allowing users logged on the legacy system to utilize some of the functionality we intend to support via our Web-based clinical counterpart. In doing so, we must distinguish between two forms of implementation. The first implementation targets context-sensitive information retrieval from Web-based resources. Providers using the legacy system can now access detailed drug monographs from a licensed drug database on the Web. This information is automatically loaded in the browser, with no intervention, as soon as the provider requests it in the emulator (Figure 2). Medline searches can now be triggered to deliver updated research on problems in the patient's problem list. Specific clinical guidelines (from the National Guidelines Clearinghouse) can be referenced by yet another selection. The second implementation targets retrieval of information that exists in the hospital information system but that was so far unavailable. An EKG server can now transmit images of patient EKGs to the browser upon requests entered in the character-based emulator. A radiology image viewer is expected to go live shortly. A Web-based clinical formula calculator uses patient lab data to provide results on the fly. These services have been independently developed as part of the original Web migration plan but can now be used, much ahead of schedule, using MBridge.

Future implementation – A third method of implementation is still in development but is worth mentioning. This method uses another component called "the coordinator" to synchronize the browser and the emulator. It relies on the architecture used in "Web tours" where "a guide" controls the browser remotely. Using the coordinator, we will be able to launch synchronous sessions, delivering clinical content to the browser (using the new Web-based information system) in parallel to the session on the legacy emulator. While the user inquires for patient lab results on the emulator, the browser will deliver the same data using the new Web-based system. This implementation is achieved by sequencing triggers to the stub program and by maintaining a joint session database in the MBridge server. Such synchronization will allow us to expose users to new functionality in a better way than the traditional email notification, while allowing users to experience it before committing to migrate.

Security model – MBridge is built to operate in close coordination with a clinical system. As such, it must use identifiers to facilitate the synchronization and information retrieval in its target applications. Any system that potentially exposes confidential patient information must be built from the ground up with appropriate interventions to prevent such breach. Such measures typically translate to an overhead in performance and introduce some rigidity in operation. Our system, targeting connectivity and responsiveness as its main goal, would be significantly disadvantaged if needed to comply with such constraints. To avoid this limitation, we made sure that identifiable clinical information would not travel via MBridge. Identifiers passed between applications (e.g. for EKG retrieval) point to resources rather than to patients. The level of security required by the target applications (e.g. the EKG viewer) is not altered and must be satisfied irrespective of the way it was initialized (typically SSL and the use of hardware tokens for identification and authentication). By keeping the functionality of the system strictly as an intra-firewall connector rather than data exchanger, we can completely bypass one of the most prevalent obstacles in clinical system development and implementation.

Discussion - Many health care organizations are now faced with the need to convert legacy clinical information systems to support Web technology. Such endeavors are undertaken in view of the advantages Web-based applications offer over traditional client server architecture. A partial list of such advantages include access to existing Web information repositories, a wider spectrum of displayable data formats, intuitive interface, decreased cost of development and deployment, enhanced interoperability and even improved security. The process of migration, however, is a demanding challenge considering Web implementations are regarded as contenders to a familiar and trusted system. During the lengthy transition process, users are required to operate both platforms as well as undergo the learning curve of the new applications. This disruption of the busy daily workflow undermines the support of the project and may alter the course and pace of its completion. The system we present in this paper allowed us to unleash the power of the Web implementation to legacy system users without forcing them to commit to the migration. We can now gradually expose the functionality in our Web system to users as they work on the legacy system without interfering with their workflow (Figure 3).

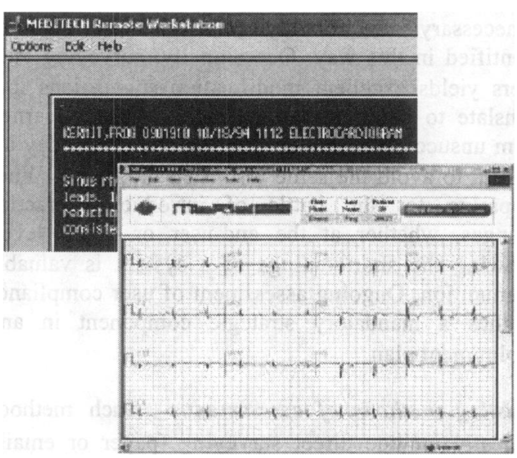

Figure 3: Exposing Images

Users and web developers in our institution welcomed this method of weaving the Web into the legacy system via a bridging architecture. The fact that the intervention requires almost no client installation or code alteration in either the legacy or Web-based systems makes it an attractive deployment from the organizational perspective. The apparent integration of context-sensitive Web functionality into the workflow makes it attractive to end- users. Finally, we believe that the architecture we present can be utilized beyond clinical system implementation to expedite the exposure of dedicated client-server users to Web technology.

References

1 McDonald CJ;Overhage JM; Dexter PR; Blevins L; Meeks-Johnson J;Suico JG; Tucker MC; Schadow G; Canopy computing: using the Web in clinical practice; JAMA 1998 Oct21;280(15):1325-9

2 Quayle C; Nothing but 'Net'. For facility managers, the Internet is a necessity--not a novelty.; HealthFacil Manage. 1999 Oct;12(10):20-4.

3 Korpela M; From legacy systems via client/server to web browser technology in hospital informatics in Finland; Medinfo 1998;9 Pt 1:222-6

4 Wang DJ; Harkness KB; Allshouse C; Elliot L; Szekalski S; Mandell SF; Development of a web based electronic patient record extending accessibility to clinical information and integrating ancillary applications; Proc AMIA Symp 1998;:131-4

5 Halamka JD, et al.; CareWeb, a web-based medical record for an integrated healthcare delivery system; Medinfo. 1998;9 Pt 1:36-9

6 Wei L; Sengupta S; Session management for web-based healthcare applications.; Proc AMIA Symp 1999;:999-1003

7 Sanders NW; Mann NH 3rd; Spengler DM; Web client and ODBC access to legacy database information: a low cost approach; Proc AMIA Annu Fall Symp 1997;:799-803

8 Lovis C; Baud RH; Scherrer JR; Internet integrated in the daily medical practice within an electronic patient record; Comput Biol Med 1998 Sep;28(5):567-79

9 Turisco F;Using Internet technology to extend access to legacy systems.; Healthc Financ Manage 1999 May;53(5):86

10 Goldberg HI; Tarczy-Hornoch P; Stephens K; Larson EB;LoGerfo JP; Internet access to patients' records; Lancet 1998 Jun 13;351(9118):1811

11 Hubbs PR, et al.; Medical information on the Internet.; JAMA. 1998 Oct 21;280(15):1363.

12 Glowniak JV; Medical resources on the Internet; Ann Intern Med 1995 Jul 15;123(2):123-31

13 Kane B, et al; Guidelines for the clinical use of electronic mail with patients. The AMIA Internet Working Group, Task Force on Guidelines for the Use of Clinic-Patient Electronic Mail; J Am Med Inform Assoc. 1998 Jan-Feb;5(1):104-11

14 Briggs JS; Early GH; Internet developments and their significance for healthcare; Med Inform Internet Med 1999 Jul-Sep;24(3):149-64

15 Finkelstein J; Cabrera MR; Hripcsak G; Internet-based home asthma telemonitoring: can patients handle the technology?; Chest 2000 Jan;117(1):148-55

16 Baldwin G; Intranets are the key to new database strategies.; Health Data Manag 1999 Nov;7(11):56-8, 60, 62-4

17 Okstein CJ; XML: a key technology for sharing clinical information; MD Comput. 1999 Sep-Oct;16(5):31-3

18 Rind DM, et al; Maintaining the confidentiality of medical records shared over the Internet and the World Wide Web; Ann Intern Med. 1997 Jul 15;127(2):138-41

19 Smith AB; Hahn AW; Bringing a WEB-based interface to a hospital information system; Biomed Sci Instrum 1997;34:303-8